12/11/2017

Warehouse Database

TO
MANAGE MANUFACTURER WAREHOUSE
USING MYSQL

Eklavya Saxena
NUID: 1850025
INFO6210
DATA MGMT AND DATABASE DESIGN

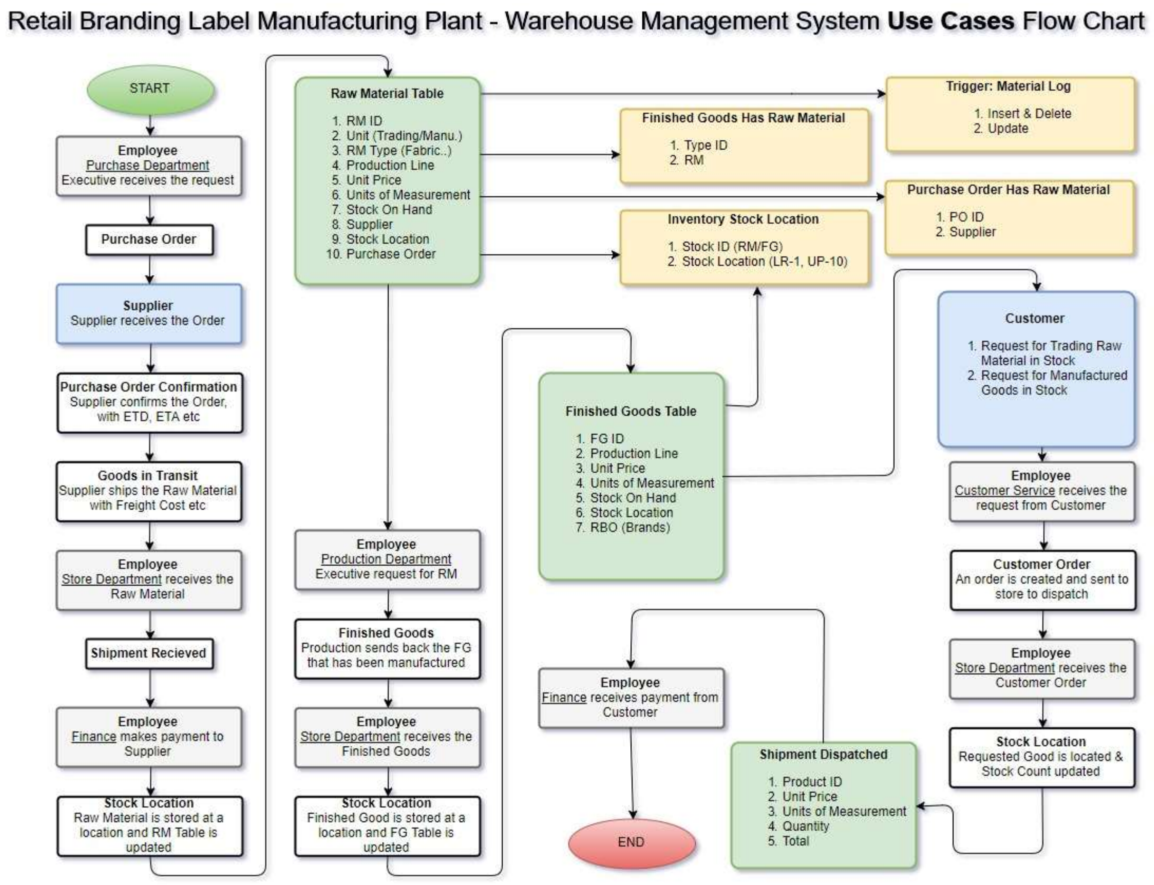
Overview

The database is customized for a Garment Label Manufacturing Plant. It is a B2B company and supplies the finished goods to garment manufacturer (customers). For manufacturing, the company buys raw material and provides it to its production plant to convert them into ready to ship finished goods. There are three different types of raw materials ordered in different colors, namely fabric, paper and RFID. There are total three production lines, namely Printed Fabric Label (PFL), Radio Frequency Identification (RFID) and Thermal Print Service Bureau (TPSB). To manage the company's inventory, following are the key departments involved:

- Purchase (raise a request for raw material),
- Store (receive the raw material and dispatch the finished goods),
- Finance (accounts receivable and accounts payable), and
- Customer Service (take a request of finished goods)

Use Cases Flow Chart

Below is the brief flow chart to understand the business done in warehouse.

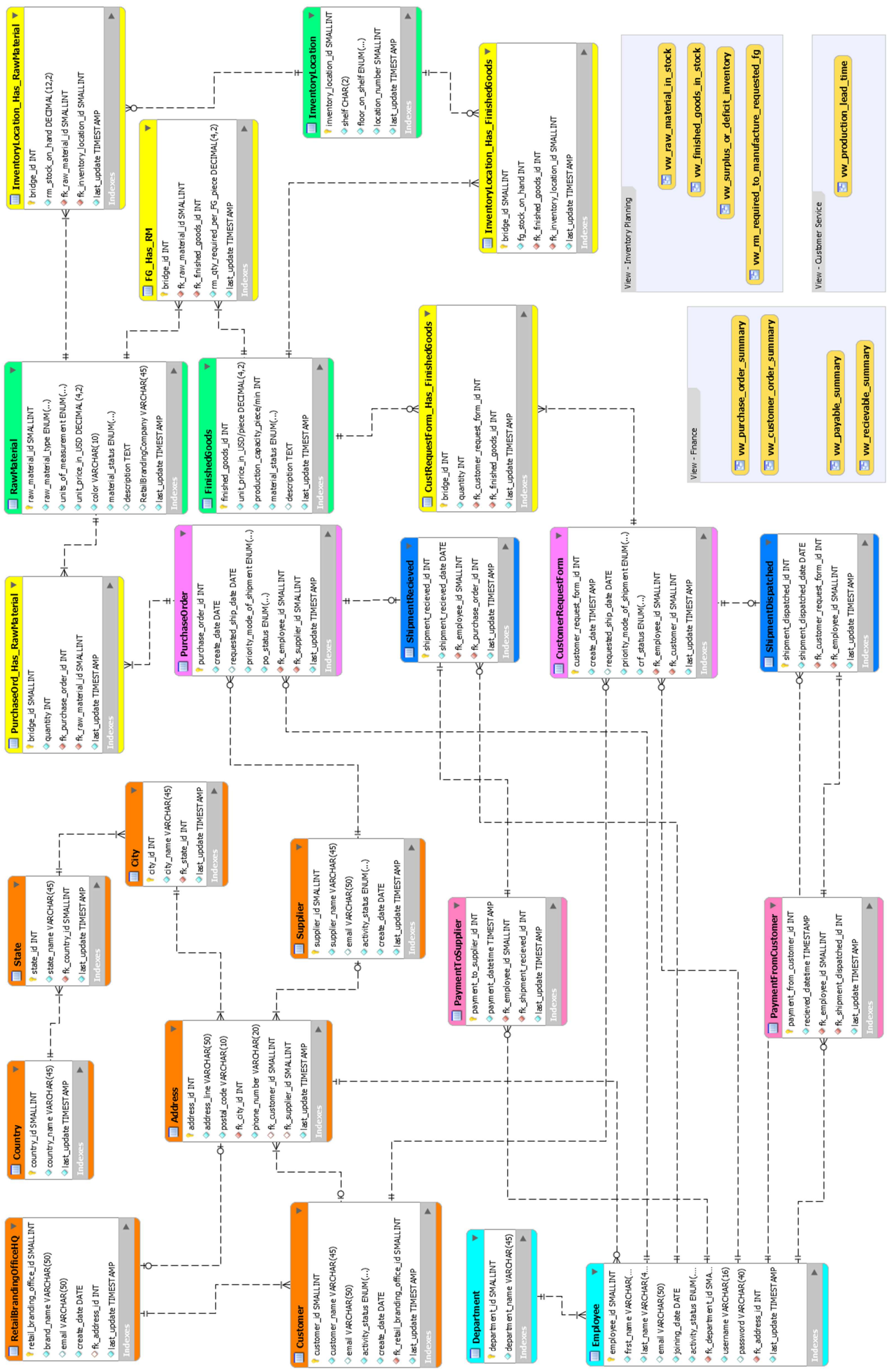

Retail Branding Label Manufacturing Plant - Warehouse Management System Use Cases Flow Chart
START
Employee
Purchase Department
Executive receives the request
Purchase Order
Supplier
Supplier receives the Order
Purchase Order Confirmation
Supplier confirms the Order, with ETD, ETA etc
Goods in Transit
Supplier ships the Raw Material with Freight Cost etc
Employee
Store Department receives the Raw Material
Shipment Recieved
Employee
Finance makes payment to Supplier
Stock Location
Raw Material is stored at a location and RM Table is updated
Raw Material Table
1. RM ID
2. Unit (Trading/Manu.)
3. RM Type (Fabric..)
4. Production Line
5. Unit Price
6. Units of Measurement
7. Stock On Hand
8. Supplier
9. Stock Location
10. Purchase Order
Employee
Production Department
Executive request for RM
Finished Goods
Production sends back the FG that has been manufactured
Employee
Store Department receives the Finished Goods
Stock Location
Finished Good is stored at a location and FG Table is updated
Finished Goods Has Raw Material
1. Type ID
2. RM
Inventory Stock Location
1. Stock ID (RM/FG)
2. Stock Location (LR-1, UP-10)
Trigger: Material Log
1. Insert & Delete
2. Update
Purchase Order Has Raw Material
1. PO ID
2. Supplier
Finished Goods Table
1. FG ID
2. Production Line
3. Unit Price
4. Units of Measurement
5. Stock On Hand
6. Stock Location
7. RBO (Brands)
Customer
1. Request for Trading Raw Material in Stock
2. Request for Manufactured Goods in Stock
Employee
Customer Service receives the request from Customer
Customer Order
An order is created and sent to store to dispatch
Employee
Store Department receives the Customer Order
Stock Location
Requested Good is located & Stock Count updated
Shipment Dispatched
1. Product ID
2. Unit Price
3. Units of Measurement
4. Quantity
5. Total
Employee
Finance receives payment from Customer
END

Purpose and Scope

The Inventory Database will:

- Provide streamlined material flow across various external and internal stakeholders.
- Enable the management to do inventory analytics (re-order stock level basis on customer's demand)
- Keep a log of the material movement for auditing purpose (issue of raw material from store).
- Maintain adequate stock levels and reduce the times store goes "Out of Stock".

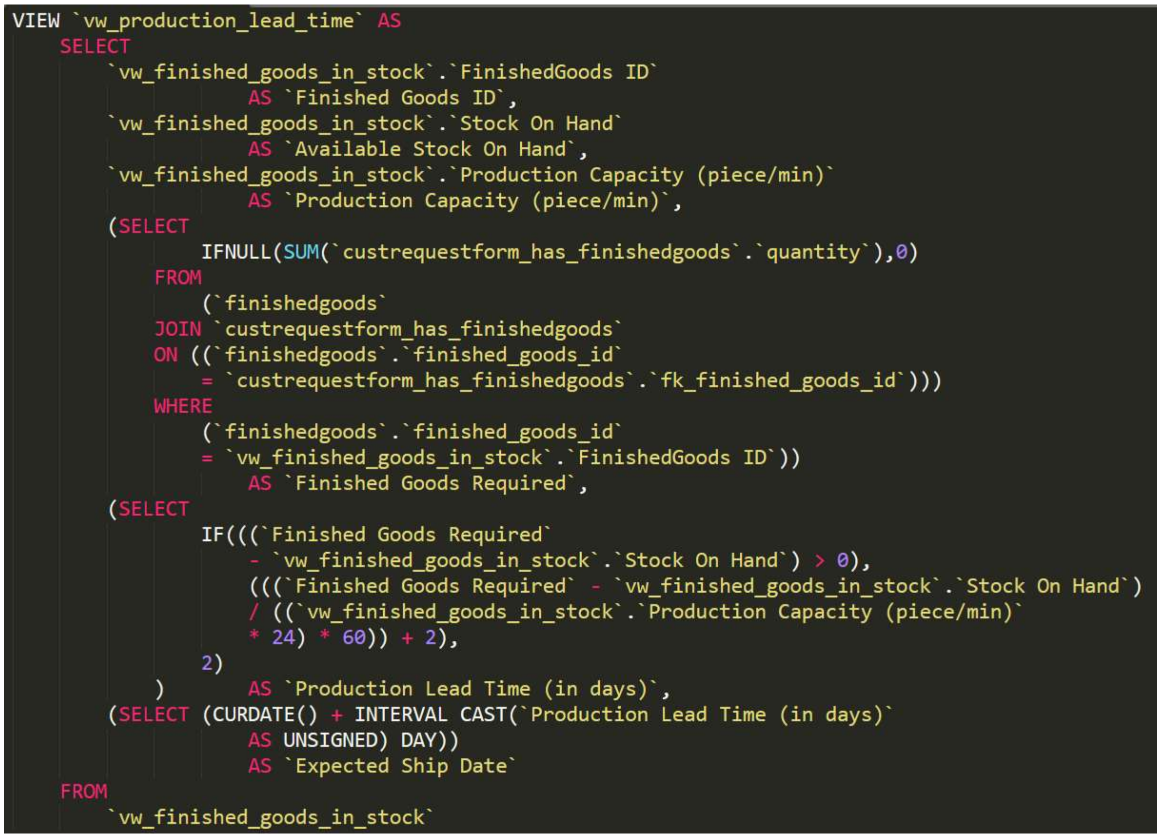
EER Diagram


InventoryLocation_Has_RawMaterial
bridge_id INT
rm_stock_on_hand DECIMAL(12,2)
fk_raw_material_id SMALLINT
fk_inventory_location_id SMALLINT
last_update TIMESTAMP
Indexes
InventoryLocation
inventory_location_id SMALLINT
shelf CHAR(2)
floor_on_shelf ENUM(...)
location_number SMALLINT
last_update TIMESTAMP
Indexes
InventoryLocation_Has_FinishedGoods
bridge_id SMALLINT
fg_stock_on_hand INT
fk_finished_goods_id INT
fk_inventory_location_id SMALLINT
last_update TIMESTAMP
Indexes
FG_Has_RM
bridge_id INT
fk_raw_material_id SMALLINT
fk_finished_goods_id INT
rm_qty_required_per_FG_piece DECIMAL(4,2)
last_update TIMESTAMP
Indexes
View - Inventory Planning
vw_raw_material_in_stock
vw_finished_goods_in_stock
vw_surplus_or_deficit_inventory
vw_rm_required_to_manufacture_requested_fg
View - Customer Service
vw_production_lead_time
View - Finance
vw_purchase_order_summary
vw_customer_order_summary
vw_payable_summary
vw_recievable_summary
RawMaterial
raw_material_id SMALLINT
raw_material_type ENUM(...)
units_of_measurement ENUM(...)
unit_price_in_USD DECIMAL(4,2)
color VARCHAR(10)
material_status ENUM(...)
description TEXT
RetailBrandingCompany VARCHAR(45)
last_update TIMESTAMP
Indexes
FinishedGoods
finished_goods_id INT
unit_price_in_USD/piece DECIMAL(4,2)
production_capacity_piece/min INT
material_status ENUM(...)
description TEXT
last_update TIMESTAMP
Indexes
CustRequestForm_Has_FinishedGoods
bridge_id INT
quantity INT
fk_customer_request_form_id INT
fk_finished_goods_id INT
last_update TIMESTAMP
Indexes
PurchaseOrd_Has_RawMaterial
bridge_id SMALLINT
quantity INT
fk_purchase_order_id INT
fk_raw_material_id SMALLINT
last_update TIMESTAMP
Indexes
PurchaseOrder
purchase_order_id INT
create_date DATE
requested_ship_date DATE
priority_mode_of_shipment ENUM(...)
po_status ENUM(...)
fk_employee_id SMALLINT
fk_supplier_id SMALLINT
last_update TIMESTAMP
Indexes
ShipmentRecieved
shipment_recieved_id INT
shipment_recieved_date DATE
fk_employee_id SMALLINT
fk_purchase_order_id INT
last_update TIMESTAMP
Indexes
CustomerRequestForm
customer_request_form_id INT
create_date TIMESTAMP
requested_ship_date DATE
priority_mode_of_shipment ENUM(...)
crf_status ENUM(...)
fk_employee_id SMALLINT
fk_customer_id SMALLINT
last_update TIMESTAMP
Indexes
ShipmentDispatched
shipment_dispatched_id INT
shipment_dispatched_date DATE
fk_customer_request_form_id INT
fk_employee_id SMALLINT
last_update TIMESTAMP
Indexes
City
city_id INT
city_name VARCHAR(45)
fk_state_id INT
last_update TIMESTAMP
Indexes
State
state_id INT
state_name VARCHAR(45)
fk_country_id SMALLINT
last_update TIMESTAMP
Indexes
Supplier
supplier_id SMALLINT
supplier_name VARCHAR(45)
email VARCHAR(50)
activity_status ENUM(...)
create_date DATE
last_update TIMESTAMP
Indexes
PaymentToSupplier
payment_to_supplier_id INT
payment_datetime TIMESTAMP
fk_employee_id SMALLINT
fk_shipment_recieved_id INT
last_update TIMESTAMP
Indexes
PaymentFromCustomer
payment_from_customer_id INT
recieved_datetime TIMESTAMP
fk_employee_id SMALLINT
fk_shipment_dispatched_id INT
last_update TIMESTAMP
Indexes
Country
country_id SMALLINT
country_name VARCHAR(45)
last_update TIMESTAMP
Indexes
Address
address_id INT
address_line VARCHAR(50)
postal_code VARCHAR(10)
fk_city_id INT
phone_number VARCHAR(20)
fk_customer_id SMALLINT
fk_supplier_id SMALLINT
last_update TIMESTAMP
Indexes
RetailBrandingOfficeHQ
retail_branding_office_id SMALLINT
brand_name VARCHAR(50)
email VARCHAR(50)
create_date DATE
fk_address_id INT
last_update TIMESTAMP
Indexes
Customer
customer_id SMALLINT
customer_name VARCHAR(45)
email VARCHAR(50)
activity_status ENUM(...)
create_date DATE
fk_retail_branding_office_id SMALLINT
last_update TIMESTAMP
Indexes
Department
department_id SMALLINT
department_name VARCHAR(45)
Indexes
Employee
employee_id SMALLINT
first_name VARCHAR(...
last_name VARCHAR(4...
email VARCHAR(50)
joining_date DATE
activity_status ENUM(...
fk_department_id SMA...
username VARCHAR(16)
password VARCHAR(40)
fk_address_id INT
last_update TIMESTAMP
Indexes

Analytics for Customer Service

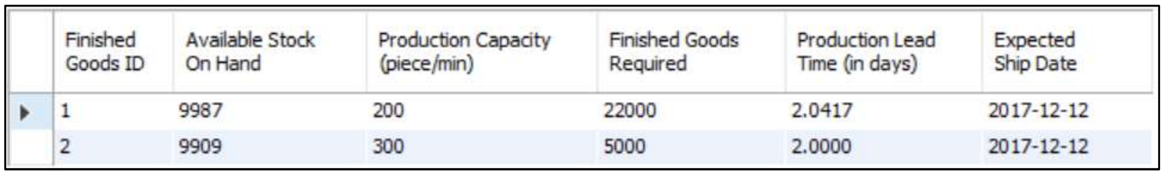
Production Lead Time & Expected Ship Date – as per Production Capacity

The view **vw_production_lead_time** shows the lead time that production will take to manufacture the deficit finished goods quantity. This enables the Customer Service to give an estimated ship date to the Customer and gain their confidence over on-time-delivery. It uses an important view **vw_rm_required_to_manufacture_requested** which calculates the total finished goods inventory on hand.



```
VIEW `vw_production_lead_time` AS
    SELECT
        `vw_finished_goods_in_stock`.`FinishedGoods ID`
                    AS `Finished Goods ID`,
        `vw_finished_goods_in_stock`.`Stock On Hand`
                    AS `Available Stock On Hand`,
        `vw_finished_goods_in_stock`.`Production Capacity (piece/min)`
                    AS `Production Capacity (piece/min)`,
        (SELECT
                IFNULL(SUM(`custrequestform_has_finishedgoods`.`quantity`),0)
            FROM
                (`finishedgoods`
            JOIN `custrequestform_has_finishedgoods`
            ON ((`finishedgoods`.`finished_goods_id`
                = `custrequestform_has_finishedgoods`.`fk_finished_goods_id`)))
            WHERE
                (`finishedgoods`.`finished_goods_id`
                = `vw_finished_goods_in_stock`.`FinishedGoods ID`))
                    AS `Finished Goods Required`,
        (SELECT
                IF(((`Finished Goods Required`
                    - `vw_finished_goods_in_stock`.`Stock On Hand`) > 0),
                    (((`Finished Goods Required` - `vw_finished_goods_in_stock`.`Stock On Hand`)
                    / ((`vw_finished_goods_in_stock`.`Production Capacity (piece/min)`
                    * 24) * 60)) + 2),
                2)
            )       AS `Production Lead Time (in days)`,
        (SELECT (CURDATE() + INTERVAL CAST(`Production Lead Time (in days)`
                    AS UNSIGNED) DAY))
                    AS `Expected Ship Date`
    FROM
        `vw_finished_goods_in_stock`
```

Output



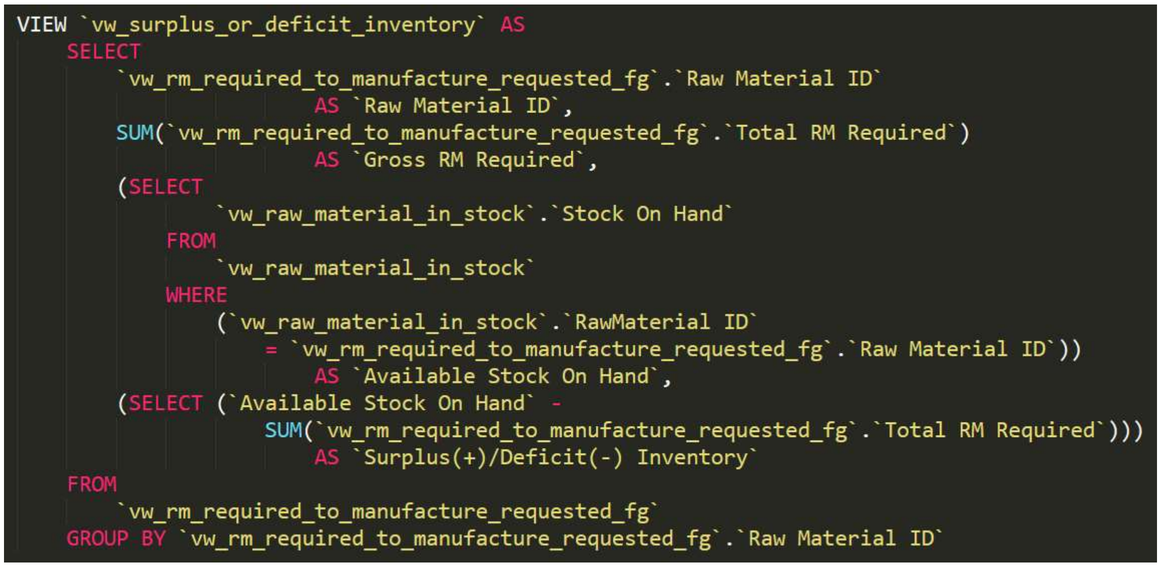
Finished Goods ID	Available Stock On Hand	Production Capacity (piece/min)	Finished Goods Required	Production Lead Time (in days)	Expected Ship Date
1	9987	200	22000	2.0417	2017-12-12
2	9909	300	5000	2.0000	2017-12-12

Analytics for Inventory Planner

Surplus or Deficit Raw Material Inventory Levels – as per Customer Orders

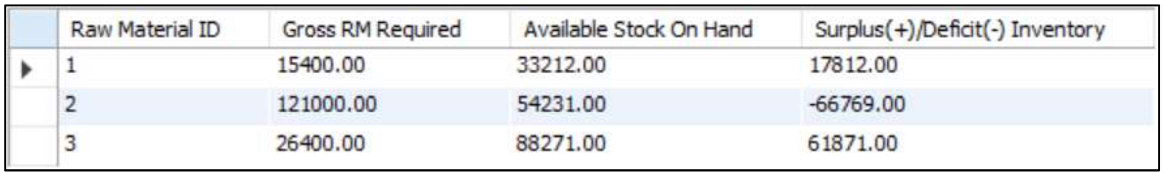
The view **vw_surplus_or_deficit_inventory** shows the raw material levels based on the customer orders on hand. This enables the Inventory Planner to plan the raw material as per customer requirements. It uses two other important views to get the desired results:

- **vw_rm_required_to_manufacture_requested_fg** (calculates the raw material required for each requested finished goods), and
- **vw_raw_material_in_stock** (calculates the total raw material inventory on hand)



```sql
VIEW `vw_surplus_or_deficit_inventory` AS
    SELECT
        `vw_rm_required_to_manufacture_requested_fg`.`Raw Material ID`
                        AS `Raw Material ID`,
        SUM(`vw_rm_required_to_manufacture_requested_fg`.`Total RM Required`)
                        AS `Gross RM Required`,
        (SELECT
                `vw_raw_material_in_stock`.`Stock On Hand`
            FROM
                `vw_raw_material_in_stock`
            WHERE
                (`vw_raw_material_in_stock`.`RawMaterial ID`
                    = `vw_rm_required_to_manufacture_requested_fg`.`Raw Material ID`))
                        AS `Available Stock On Hand`,
        (SELECT (`Available Stock On Hand` -
                    SUM(`vw_rm_required_to_manufacture_requested_fg`.`Total RM Required`)))
                        AS `Surplus(+)/Deficit(-) Inventory`
    FROM
        `vw_rm_required_to_manufacture_requested_fg`
    GROUP BY `vw_rm_required_to_manufacture_requested_fg`.`Raw Material ID`
```

Output

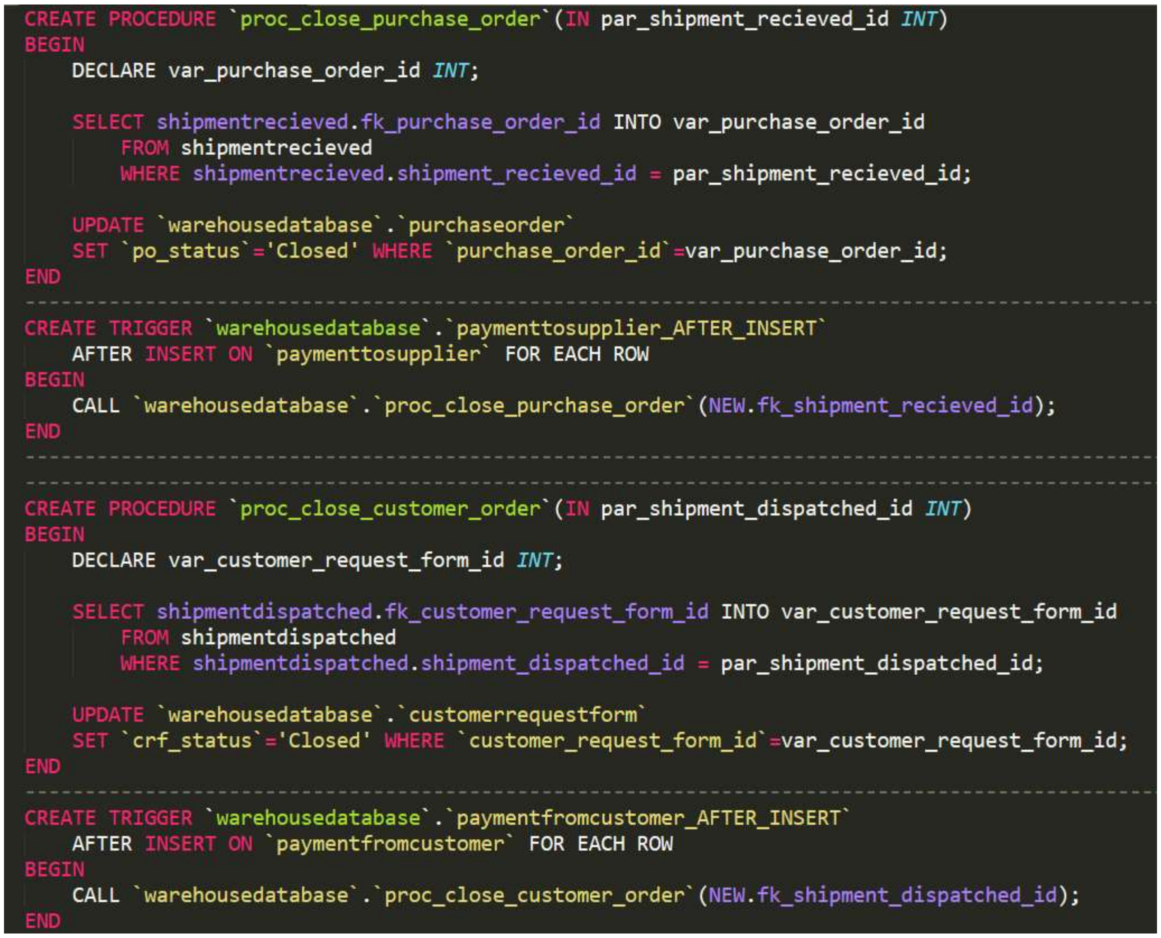


Raw Material ID	Gross RM Required	Available Stock On Hand	Surplus(+)/Deficit(-) Inventory
1	15400.00	33212.00	17812.00
2	121000.00	54231.00	-66769.00
3	26400.00	88271.00	61871.00

Restricted Auto Updates for Accounts Payable & Receivable

Updates Status of Orders using Stored Procedure & Triggers

On Insert, the Triggers on PaymentToSupplier and PaymentToCustomer tables, changes the order status of PurchaseOrder and CustomerRequestForm tables using Stored Procedures **proc_close_purchase_order** and **proc_close_customer_order** respectively, from Open to Close. This helps Finance to keep track and hold integrity of their Accounts Payable and Receivable.

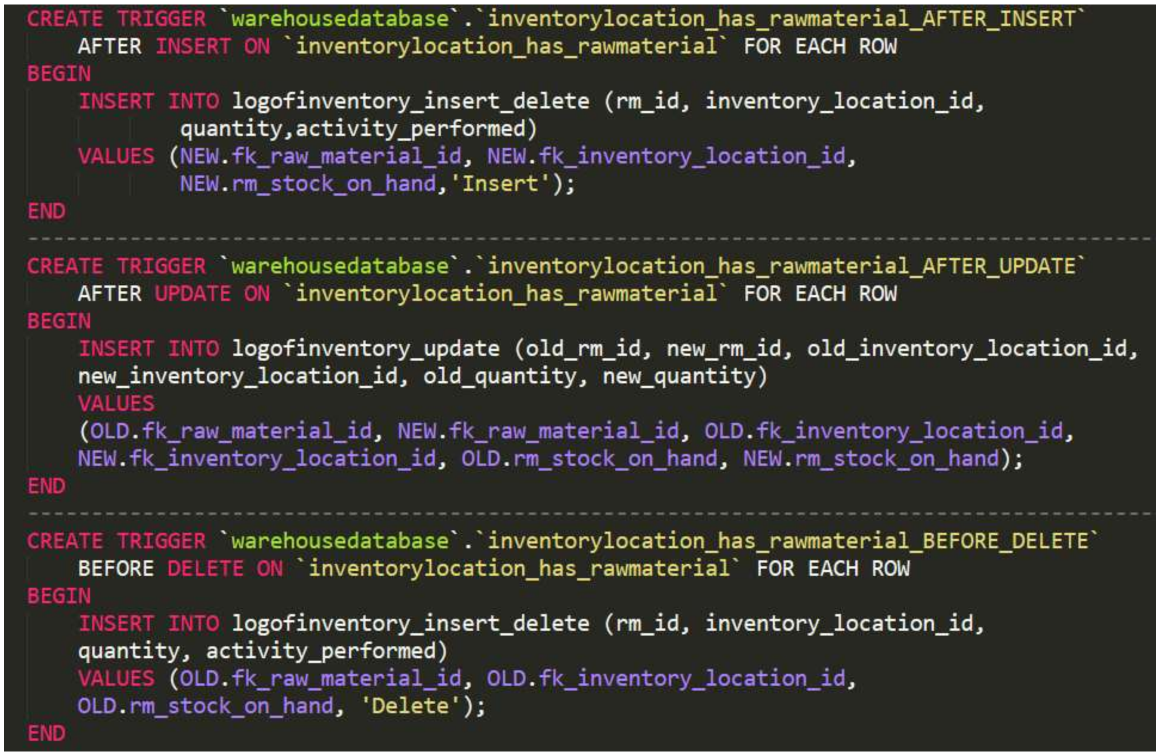


```
CREATE PROCEDURE `proc_close_purchase_order`(IN par_shipment_recieved_id INT)
BEGIN
    DECLARE var_purchase_order_id INT;

    SELECT shipmentrecieved.fk_purchase_order_id INTO var_purchase_order_id
        FROM shipmentrecieved
        WHERE shipmentrecieved.shipment_recieved_id = par_shipment_recieved_id;

    UPDATE `warehousedatabase`.`purchaseorder`
    SET `po_status`='Closed' WHERE `purchase_order_id`=var_purchase_order_id;
END
-------------------------------------------------------------------------------------------
CREATE TRIGGER `warehousedatabase`.`paymenttosupplier_AFTER_INSERT`
    AFTER INSERT ON `paymenttosupplier` FOR EACH ROW
BEGIN
    CALL `warehousedatabase`.`proc_close_purchase_order`(NEW.fk_shipment_recieved_id);
END
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
CREATE PROCEDURE `proc_close_customer_order`(IN par_shipment_dispatched_id INT)
BEGIN
    DECLARE var_customer_request_form_id INT;

    SELECT shipmentdispatched.fk_customer_request_form_id INTO var_customer_request_form_id
        FROM shipmentdispatched
        WHERE shipmentdispatched.shipment_dispatched_id = par_shipment_dispatched_id;

    UPDATE `warehousedatabase`.`customerrequestform`
    SET `crf_status`='Closed' WHERE `customer_request_form_id`=var_customer_request_form_id;
END
-------------------------------------------------------------------------------------------
CREATE TRIGGER `warehousedatabase`.`paymentfromcustomer_AFTER_INSERT`
    AFTER INSERT ON `paymentfromcustomer` FOR EACH ROW
BEGIN
    CALL `warehousedatabase`.`proc_close_customer_order`(NEW.fk_shipment_dispatched_id);
END
```

Inventory Log for Audit Purposes

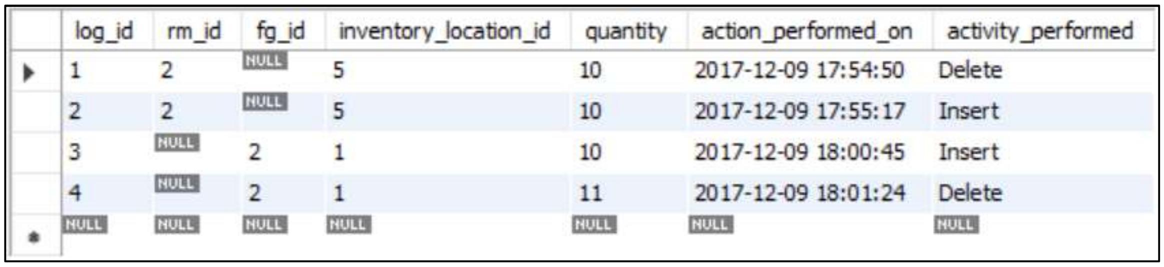
Maintain Log of Material Movement – Raw Material and Finished Goods

For a company, inventory has a huge contribution towards its assets and it becomes necessary to track the material movement. There are 3 Triggers each for Insert, Update and Delete in InventoryLocation_Has_FinishedGoods and InventoryLocation_Has_RawMaterial tables to ensure that the material movement is logged in LogOfInventory_Insert_Delete and LogOfInventory_Update tables.



```
CREATE TRIGGER `warehousedatabase`.`inventorylocation_has_rawmaterial_AFTER_INSERT`
    AFTER INSERT ON `inventorylocation_has_rawmaterial` FOR EACH ROW
BEGIN
    INSERT INTO logofinventory_insert_delete (rm_id, inventory_location_id,
            quantity,activity_performed)
    VALUES (NEW.fk_raw_material_id, NEW.fk_inventory_location_id,
            NEW.rm_stock_on_hand,'Insert');
END
------------------------------------------------------------------------------------
CREATE TRIGGER `warehousedatabase`.`inventorylocation_has_rawmaterial_AFTER_UPDATE`
    AFTER UPDATE ON `inventorylocation_has_rawmaterial` FOR EACH ROW
BEGIN
    INSERT INTO logofinventory_update (old_rm_id, new_rm_id, old_inventory_location_id,
    new_inventory_location_id, old_quantity, new_quantity)
    VALUES
    (OLD.fk_raw_material_id, NEW.fk_raw_material_id, OLD.fk_inventory_location_id,
    NEW.fk_inventory_location_id, OLD.rm_stock_on_hand, NEW.rm_stock_on_hand);
END
------------------------------------------------------------------------------------
CREATE TRIGGER `warehousedatabase`.`inventorylocation_has_rawmaterial_BEFORE_DELETE`
    BEFORE DELETE ON `inventorylocation_has_rawmaterial` FOR EACH ROW
BEGIN
    INSERT INTO logofinventory_insert_delete (rm_id, inventory_location_id,
    quantity, activity_performed)
    VALUES (OLD.fk_raw_material_id, OLD.fk_inventory_location_id,
    OLD.rm_stock_on_hand, 'Delete');
END
```

Output

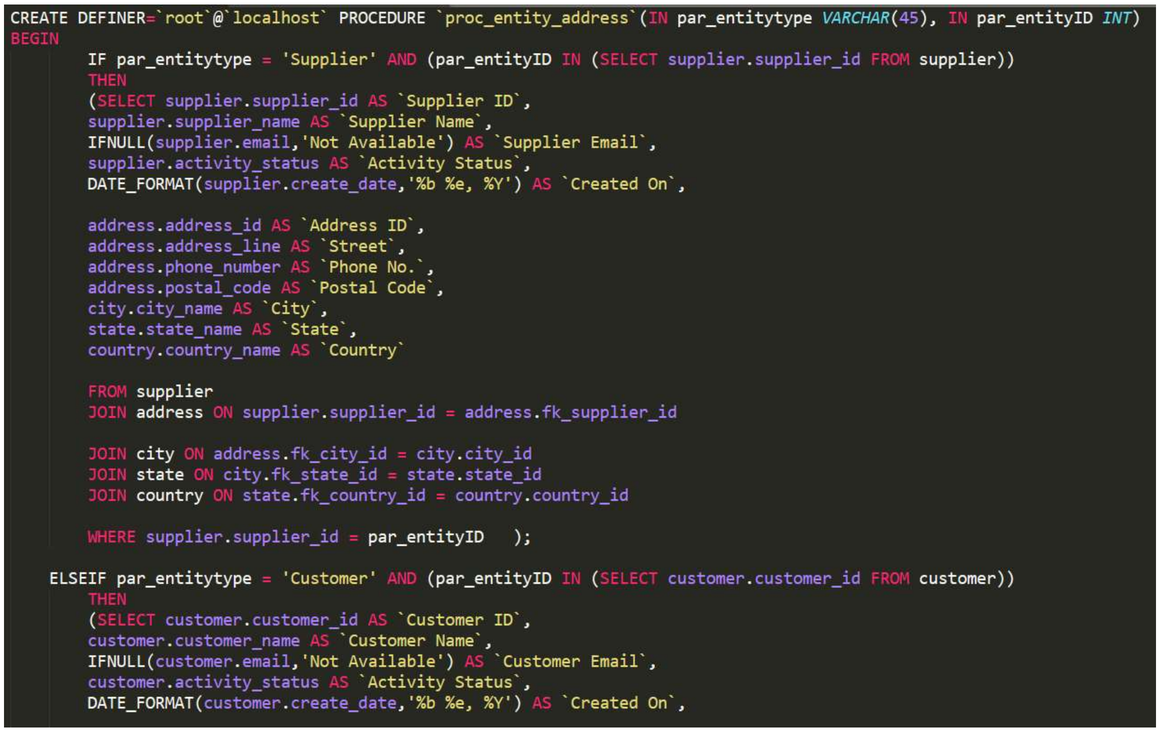


log_id	rm_id	fg_id	inventory_location_id	quantity	action_performed_on	activity_performed
1	2	NULL	5	10	2017-12-09 17:54:50	Delete
2	2	NULL	5	10	2017-12-09 17:55:17	Insert
3	NULL	2	1	10	2017-12-09 18:00:45	Insert
4	NULL	2	1	11	2017-12-09 18:01:24	Delete
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Address of Related Entities

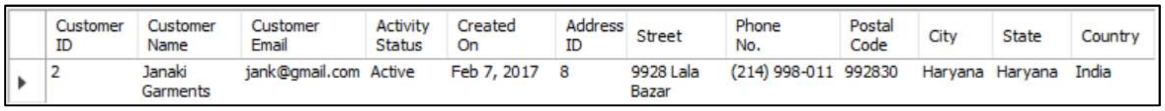
Stored Procedure – Use to retrieve address of any entity related to the Company

A common stored procedure is created to retrieve address of entity using its type (e.g. Supplier, Employee etc.) and its ID. Also, if the data is not found in the database, exception handling ERROR 1644 has been taken care of at database level.



```
CREATE DEFINER=`root`@`localhost` PROCEDURE `proc_entity_address`(IN par_entitytype VARCHAR(45), IN par_entityID INT)
BEGIN
        IF par_entitytype = 'Supplier' AND (par_entityID IN (SELECT supplier.supplier_id FROM supplier))
        THEN
        (SELECT supplier.supplier_id AS `Supplier ID`,
        supplier.supplier_name AS `Supplier Name`,
        IFNULL(supplier.email,'Not Available') AS `Supplier Email`,
        supplier.activity_status AS `Activity Status`,
        DATE_FORMAT(supplier.create_date,'%b %e, %Y') AS `Created On`,

        address.address_id AS `Address ID`,
        address.address_line AS `Street`,
        address.phone_number AS `Phone No.`,
        address.postal_code AS `Postal Code`,
        city.city_name AS `City`,
        state.state_name AS `State`,
        country.country_name AS `Country`

        FROM supplier
        JOIN address ON supplier.supplier_id = address.fk_supplier_id

        JOIN city ON address.fk_city_id = city.city_id
        JOIN state ON city.fk_state_id = state.state_id
        JOIN country ON state.fk_country_id = country.country_id

        WHERE supplier.supplier_id = par_entityID   );

    ELSEIF par_entitytype = 'Customer' AND (par_entityID IN (SELECT customer.customer_id FROM customer))
        THEN
        (SELECT customer.customer_id AS `Customer ID`,
        customer.customer_name AS `Customer Name`,
        IFNULL(customer.email,'Not Available') AS `Customer Email`,
        customer.activity_status AS `Activity Status`,
        DATE_FORMAT(customer.create_date,'%b %e, %Y') AS `Created On`,
```

Output

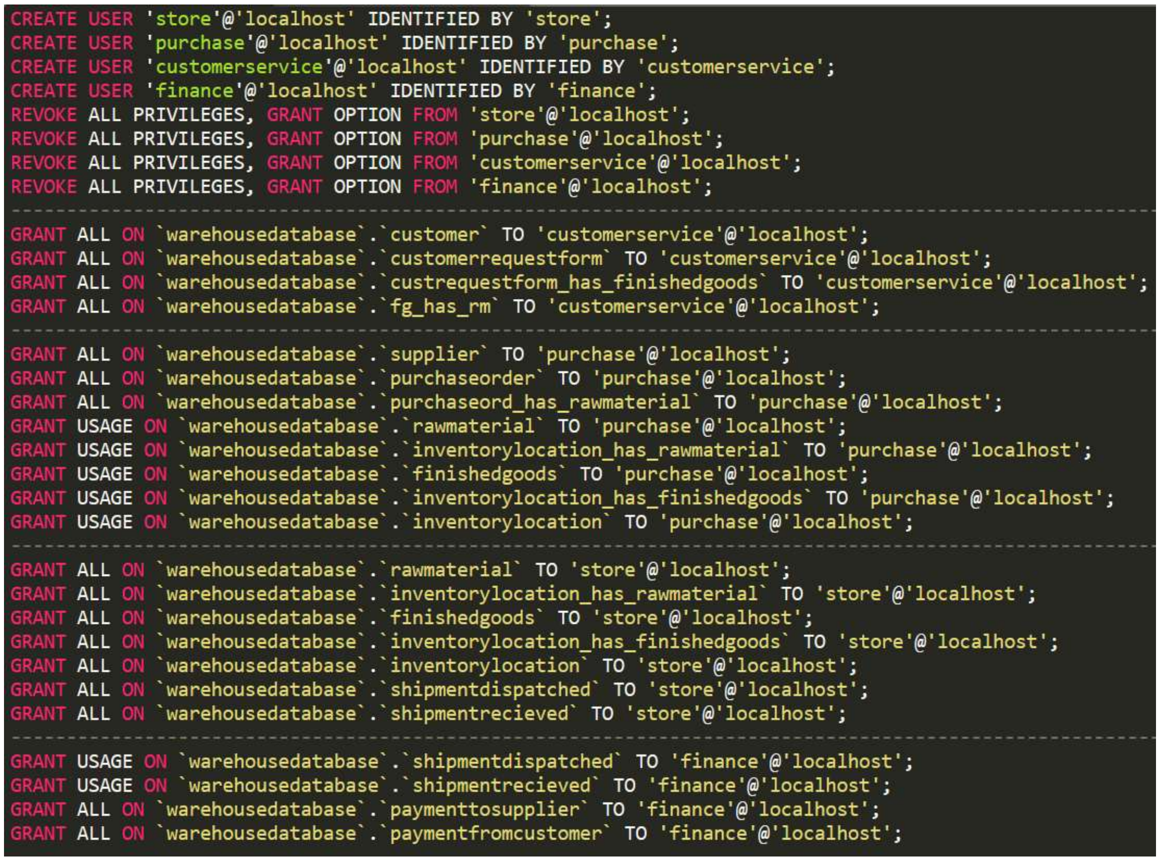


Customer ID	Customer Name	Customer Email	Activity Status	Created On	Address ID	Street	Phone No.	Postal Code	City	State	Country
2	Janaki Garments	jank@gmail.com	Active	Feb 7, 2017	8	9928 Lala Bazar	(214) 998-011	992830	Haryana	Haryana	India

Users and Privileges

Grants to Access Database Tables

It is important to restrict employee from different department to access the data which is irrelevant to them. Privileges are given as per department requirement.



```
CREATE USER 'store'@'localhost' IDENTIFIED BY 'store';
CREATE USER 'purchase'@'localhost' IDENTIFIED BY 'purchase';
CREATE USER 'customerservice'@'localhost' IDENTIFIED BY 'customerservice';
CREATE USER 'finance'@'localhost' IDENTIFIED BY 'finance';
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'store'@'localhost';
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'purchase'@'localhost';
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'customerservice'@'localhost';
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'finance'@'localhost';
--------------------------------------------------------------------------------
GRANT ALL ON `warehousedatabase`.`customer` TO 'customerservice'@'localhost';
GRANT ALL ON `warehousedatabase`.`customerrequestform` TO 'customerservice'@'localhost';
GRANT ALL ON `warehousedatabase`.`custrequestform_has_finishedgoods` TO 'customerservice'@'localhost';
GRANT ALL ON `warehousedatabase`.`fg_has_rm` TO 'customerservice'@'localhost';
--------------------------------------------------------------------------------
GRANT ALL ON `warehousedatabase`.`supplier` TO 'purchase'@'localhost';
GRANT ALL ON `warehousedatabase`.`purchaseorder` TO 'purchase'@'localhost';
GRANT ALL ON `warehousedatabase`.`purchaseord_has_rawmaterial` TO 'purchase'@'localhost';
GRANT USAGE ON `warehousedatabase`.`rawmaterial` TO 'purchase'@'localhost';
GRANT USAGE ON `warehousedatabase`.`inventorylocation_has_rawmaterial` TO 'purchase'@'localhost';
GRANT USAGE ON `warehousedatabase`.`finishedgoods` TO 'purchase'@'localhost';
GRANT USAGE ON `warehousedatabase`.`inventorylocation_has_finishedgoods` TO 'purchase'@'localhost';
GRANT USAGE ON `warehousedatabase`.`inventorylocation` TO 'purchase'@'localhost';
--------------------------------------------------------------------------------
GRANT ALL ON `warehousedatabase`.`rawmaterial` TO 'store'@'localhost';
GRANT ALL ON `warehousedatabase`.`inventorylocation_has_rawmaterial` TO 'store'@'localhost';
GRANT ALL ON `warehousedatabase`.`finishedgoods` TO 'store'@'localhost';
GRANT ALL ON `warehousedatabase`.`inventorylocation_has_finishedgoods` TO 'store'@'localhost';
GRANT ALL ON `warehousedatabase`.`inventorylocation` TO 'store'@'localhost';
GRANT ALL ON `warehousedatabase`.`shipmentdispatched` TO 'store'@'localhost';
GRANT ALL ON `warehousedatabase`.`shipmentrecieved` TO 'store'@'localhost';
--------------------------------------------------------------------------------
GRANT USAGE ON `warehousedatabase`.`shipmentdispatched` TO 'finance'@'localhost';
GRANT USAGE ON `warehousedatabase`.`shipmentrecieved` TO 'finance'@'localhost';
GRANT ALL ON `warehousedatabase`.`paymenttosupplier` TO 'finance'@'localhost';
GRANT ALL ON `warehousedatabase`.`paymentfromcustomer` TO 'finance'@'localhost';
```